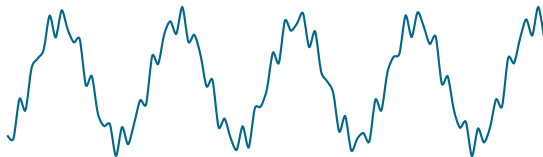


# Connecting Musical Modules

## Musical Hardware and Software Interfaces

Joren Six

*IPEM, University Ghent*



March 19, 2016

# OVERVIEW

## INTRODUCTION

## Hardware modules

### Axoloti

Axoloti demo

### Teensy

USB Device Types

Teensy demo

## Software modules

### TarsosDSP

TarsosDSP demo

### Android

Android Demo

### Patcher

## Protocols

### MIDI

### OSC

# INTRODUCTION

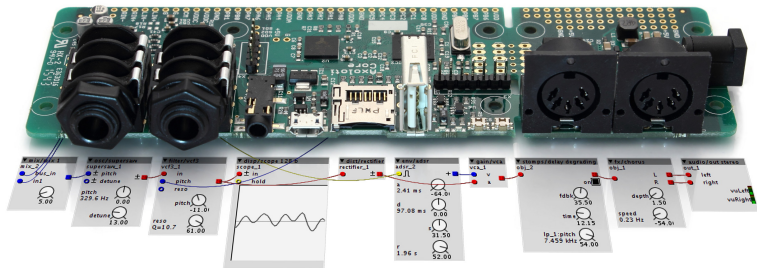
## Goal

Give a practical overview of hard- and software components for DIY musical applications and how to make them communicate.

# INTRODUCTION

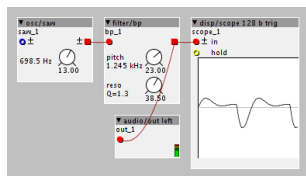
- ▶ Hardware components
- ▶ Software systems
- ▶ Protocols

# AXOLOTI



**Fig:** Axoloti, a digital audio platform for makers.

# AXOLOTI



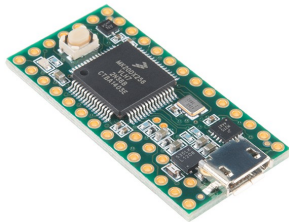
**Fig:** The Axoloti Patcher environment is used to program it.

- ▶ Stereo audio input and output.
- ▶ Audio at 16 bit, 44.1 kHz sample rate
- ▶ SD-card slot for audio samples
- ▶ Crowd-funded on indigogo
- ▶ Designed by Johannes Taelman

# AXOLOTI DEMO

Axoloti reacts to scale sensor.

# TEENSY



**Fig:** Teensy 3.2 an Arduino compatible micro-controller.

- ▶ Arduino compatible
  - ▶ Programmable in C
  - ▶ Massive community
  - ▶ Many digital and analog sensors with software support
  - ▶ Access to software libraries
- ▶ Native (fast) USB interface
- ▶ Analog input (13 bits) and output (12 bits)
- ▶ Switches easily between USB device types
- ▶ Cheap 23 €.



# TEENSY



- ▶ Stereo audio input and output.
- ▶ Audio at 16 bit, 44.1 kHz sample rate
- ▶ SD-card slot for audio samples

**Fig:** The Audio Adapter Board for Teensy adds audio capabilities to a Teensy

# TEENSY VS ARDUINO

A classical Arduino Uno is not well suited for (high quality) audio applications.

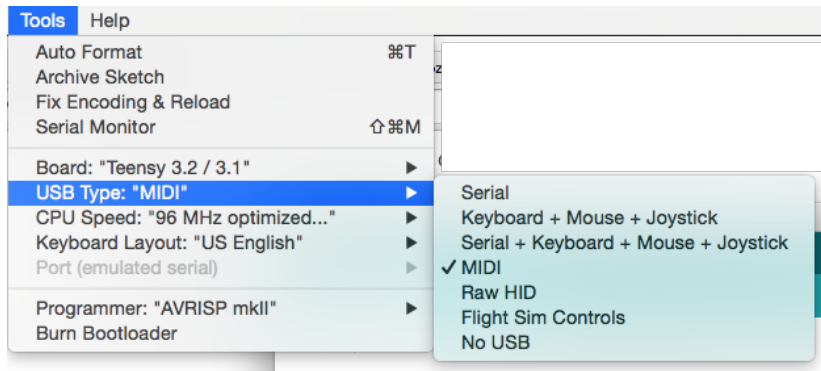
	<b>Arduino Uno</b>	<b>Teensy 3.2</b>
<b>Processor</b>	16 MHz	Cortex-M4 96 MHz
<b>Audio output</b>	8 bit analog out	12 bit or 16 bit <sup>1</sup>
<b>Memory</b>	2 kB	64kB
<b>USB Device type</b>	Configurable	Easily configurable

Table: Comparison of features between Aduino Uno and Teensy 3.2 with musical applications in mind.

---

<sup>1</sup>With the Teensy Audio Adapter

# USB DEVICE TYPES



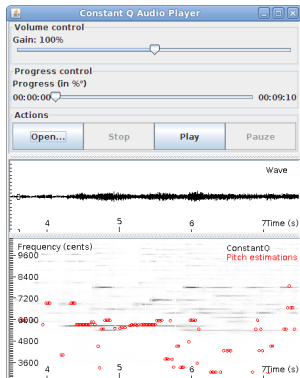
**Fig:** The Teensy device type can be selected easily, great for interaction with other hardware.

- ▶ MIDI device
- ▶ Mouse / keyboard
- ▶ Serial device

# TEENSY DEMO

Teensy does pitch detection, sends MIDI to a synthesizer.

# TARSOSDSP



**Fig:** TarsosDSP an audio processing library in Java

- ▶ Platform independent without any external dependencies
- ▶ Pitch detection algorithms, time stretching, ...
- ▶ Created by me and presented at newline 2011

# TARSOSDSP DEMO

UtterAsterisk

# ANDROID



**Fig:** Android 6.0 includes  
MIDI Support

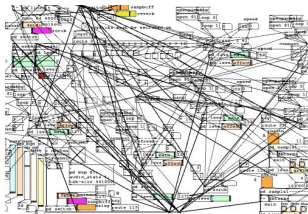
- ▶ Improved latency especially compared with 4.0
- ▶ MIDI support since 6.0
- ▶ USB-OTG allows connecting Arduino's

# ANDROID DEMO

Android MIDI keyboard



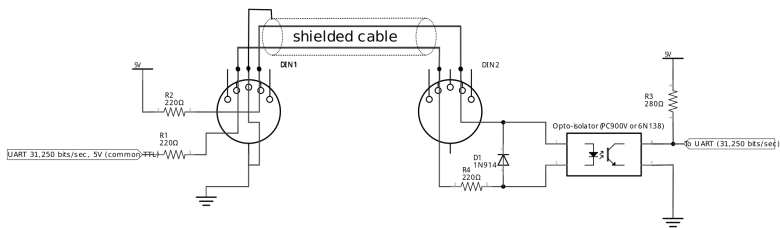
# PATCHER ENVIRONMENTS (PURE DATA - MAX/MSP)



**Fig:** A pure data patch

- ▶ Visual Programming languages
- ▶ Mainly for Music applications
- ▶ Many available objects

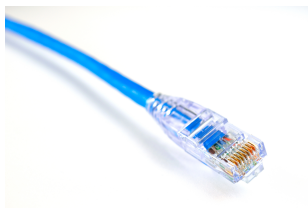
# MIDI



**Fig:** MIDI port schema

- ▶ Small messages
- ▶ Mainly over USB or serial cable
- ▶ Easy to understand
- ▶ Well supported

# OSC



**Fig:** A random Ethernet cable.

- ▶ Larger messages allowed
- ▶ Mainly used over UDP, Ethernet
- ▶ Easy to understand
- ▶ Well supported
- ▶ Ideal for inter process communication
- ▶ Can be prone to jitter

# BONUS DEMO

Auto-tune. Teensy does pitch detection, Android sends MIDI keys. TarsosDSP does pitch real-time pitch shifting based on that.